Amendments to the Specification:

1. Please insert the following new paragraph before line 3 on page 10, which begins with "Other structures . . . ."

Individual IP vendors can directly license their IP modules to PLD users under their own chosen conditions. Individual IP vendors can use time-based licenses or use-based licenses. Each PLD is manufactured with a non-volatile non-rewritable unique device identifier (UDI) that uniquely identifies the PLD. If, for example, a user obtains a license to use a particular IP module on a particular PLD, then the IP vendor issues the user an authorization code that contains the UDI (in encrypted form) of the authorized target PLD. When the user wishes to use the IP module, the user supplies the authorization code to a license manager. The license manager decrypts the authorization code and checks that the UDI of the supplied authorization code matches the UDI of the to-be-programmed PLD. If the two match, then the license manager encrypts a key for the IP module using the UDI and a private key, and then sends the encrypted key to the target PLD. The target PLD uses its UDI and the private key to decrypt the key. The key is then stored in on the PLD. When the configuration bitstream for the design that incorporates the IP module is later sent to the PLD, the license manager encrypts the IP module portion with the key. The PLD receives the bitstream and uses the stored key to decrypt the IP module portion of the bitstream. The unencrypted bitstream is then used to configure the PLD.

2. Please amend the paragraph on page 10, lines 19-20 as follows.

~~Figure 4 is~~ Figures 4A and 4B are a simplified flow-chart of a method carried out by the system 100 of Figure 2.

3. Please amend the paragraph starting at page 13, line 24, and continuing to page 14, line 2, as follows.

IP vendor 113 queries user 108 for the UDI **[[**114**]]** 116 of the target FPGA 102. In one embodiment, the user 108 uses the development system 104 to read the UDI from the target FPGA 102. In this embodiment, development system 104 includes associated interface hardware (not shown) and this interface hardware reads the UDI 116 out of FPGA 102. The interface hardware is provided with the development system so that the development system 104 can read from and/or write to FPGA 102. The user may, however, obtain the UDI 116 by means other than such interface hardware. The UDI may, for example, simply be printed on the FPGA 102.

4. Please amend the paragraph starting at page 18, line 24, and continuing to page 19, line 17, as follows.

Next, FPGA 102 receives the bitstream 101 via the FPGA's DIN terminal and then uses encryptor/decryptor 124 to decrypt (step 209) each encrypted part with its respective key. For example, when FPGA 102 receives the part of the bitstream 101 corresponding to IP module 301, it receives the start code 305 and then the key number for key 120. This key number is "one". FPGA 102 uses this key number "one" to retrieve the key 120 stored in association with key number "one" in the one-time writable non-volatile ROM. Key 120 is supplied to encryptor/decryptor 124 to decrypt the following configuration data for IP module 301. In this way, the encrypted configuration data for each IP module is decrypted using the

correct key identified by the preceding key number in the bitstream. The configuration data from the resulting decrypted bit stream 126 is loaded (step 209) into appropriate memory cells of the CLBs ~~109~~ 4, IOBs ~~111~~ 6, and configurable interconnect structure ~~110~~ 5 so as to configure FPGA 102 to realize the user-specific circuit. Because FPGA 102 in this example is a Virtex family Xilinx FPGA, the resulting decrypted bitstream 126 comports with the standard Xilinx bit stream protocol for configuring a Virtex family FPGA. If, on the other hand, the PLD being configured is not a Xilinx Virtex FPGA and therefore requires a different configuration bitstream, then decrypted bitstream 126 would comport with the protocol required by that PLD.

5. Please amend the paragraph at page 20, lines 3-20 as follows.

It is desired that one user not be able to understand or copy the design of another user by copying the bit stream loaded into the FPGA on power-up. In the above-described embodiment, the user provides key "KEY 4" 123 and this key is used to encrypt the portions of the bitstream associated with user's portion 304 of the user-specific circuit. A second user cannot therefore decipher the first user's design by examining the bitstream. The second user cannot copy the bitstream and program other FPGAs, even if the second user were to arrange with the necessary IP vendors to use the needed IP modules, because the second user would not know "KEY 4" of the first user. Because each FPGA programmed must first be loaded with the set of keys **[[**119**]]** from the license manager, and because the bitstream by which these keys is loaded is not the same FPGA to FPGA, copying the bitstream by which the keys are loaded into one FPGA would not enable the second user to load the keys into another FPGA.